(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 383 314 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
08.11.2006 Bulletin 2006/45

(43) Date of publication A2:
21.01.2004 Bulletin 2004/04

(21) Application number: 03254060.1

(22) Date of filing: 26.06.2003

(51) Int Cl.:
H04N 5/265 (2006.01) H04N 5/275 (2006.01)

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR
Designated Extension States:
AL LT LV MK

(30) Priority: 17.07.2002 GB 0216685

(71) Applicant: SONY UNITED KINGDOM LIMITED
Weybridge KT13 0XW (GB)

(72) Inventor: Witt, Sarah Elizabeth
Basingstoke,
Hampshire RG23 8WL (GB)

(74) Representative: Turner, James Arthur et al
D Young & Co
120 Holborn
London EC1N 2DY (GB)

(54) Video signal processing

(57) A video processing method generates output images for display, each output image having one or more regions derived from a first source image separated by a wipe boundary from one or more regions derived from a second source image, each display position of the source images having an associated transparency coefficient. The method comprises the steps of: preparing the first source image for display in a display memory; defining a wipe origin locus representing a wipe boundary when substantially all of the output image is derived from the first source image and a wipe destination locus representing a wipe boundary when substantially all of the output image is derived from the second source image, points on the wipe destination locus being associated with points on the wipe origin locus, each such pair of associated points defining a respective wipe progression direction; generating a wipe control signal defining proportions of the first and second source images to be displayed with respect to normalised display distance along a wipe progression direction from the wipe origin locus and the wipe destination locus; modifying the transparency coefficient of the first source image held in the display memory, the transparency coefficient of each display position being modified in dependence on value of the wipe control signal corresponding to the normalised display distance along the wipe progression direction between that display position and the wipe origin locus and between that display position and the wipe destination locus; and writing the second source image over the first source image in the display memory so that the first source image is modified by pixels of the second source image in dependence on the transparency coefficient associated with each display position of the first source image.

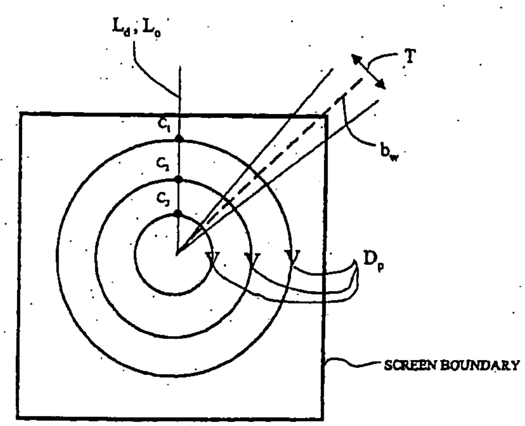

FIGURE 14

EP 1 383 314 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 25 4060

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	US 6 366 286 B1 (HERMANSON DAVID W) 2 April 2002 (2002-04-02) * column 10, line 37 - column 11, line 2 * * column 7, line 20 - line 63 * * claim 1 * -----	1-12	INV. H04N5/265 H04N5/275
X	EP 0 438 061 A (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD) 24 July 1991 (1991-07-24) * column 1, line 38 - column 2, line 58 * * column 7, line 11 - column 8, line 51 * * column 18, line 20 - column 24, line 23 * * claims 1-8 * -----	1-12	
X	US 4 954 898 A (NAKATA ET AL) 4 September 1990 (1990-09-04) * column 7, line 1 - column 9, line 23 * -----	1-12	
P,X	US 6 563 497 B1 (EBIHARA HITOSHI) 13 May 2003 (2003-05-13) * column 3, line 1 - line 67 * * column 12, line 6 - line 57 * -----	1-12	TECHNICAL FIELDS SEARCHED (IPC)
A	GB 2 348 763 A (* SONY UNITED KINGDOM LIMITED) 11 October 2000 (2000-10-11) * the whole document * -----	1-12	H04N G11B G06T
A	US 6 043 852 A (NAKATA ET AL) 28 March 2000 (2000-03-28) * column 5, line 43 - line 64 * -----	1-12	
A	US 5 235 423 A (DUNBAR ET AL) 10 August 1993 (1993-08-10) * column 6, line 3 - line 39 * ----- -/--	1,2	

The present search report has been drawn up for all claims

2

Place of search	Date of completion of the search	Examiner
The Hague	29 September 2006	Deltorn, Jean-Marc

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 03 25 4060

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	KORT A ED - SPENCER S N (ED) ASSOCIATION FOR COMPUTING MACHINERY: "COMPUTER AIDED INBETWEENING" NPAR 2002. SYMPOSIUM ON NON - PHOTOREALISTIC ANIMATION AND RENDERING. ANNECY, FRANCE, JUNE 3 - 5, 2002, SYMPOSIUM ON NON - PHOTOREALISTIC ANIMATION AND RENDERING, NEW YORK, NY : ACM, US, 3 June 2002 (2002-06-03), pages 125-132, XP001201019 ISBN: 1-58113-494-0 section 3 -----	1,8,12	

TECHNICAL FIELDS SEARCHED (IPC)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	29 September 2006	Deltorn, Jean-Marc

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

2

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO.

EP 03 25 4060

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29-09-2006

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 6366286	B1	02-04-2002	CA	2267537 A1	03-10-1999
			GB	2336264 A	13-10-1999
EP 0438061	A	24-07-1991	DE	69114312 D1	14-12-1995
			DE	69114312 T2	13-06-1996
			US	5184222 A	02-02-1993
US 4954898	A	04-09-1990	DE	3915037 A1	16-11-1989
			FR	2631189 A1	10-11-1989
			GB	2219170 A	29-11-1989
			KR	143421 B1	15-07-1998
US 6563497	B1	13-05-2003	GB	2339358 A	19-01-2000
			JP	11331700 A	30-11-1999
GB 2348763	A	11-10-2000	JP	2000324393 A	24-11-2000
US 6043852	A	28-03-2000	NONE		
US 5235423	A	10-08-1993	NONE		

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

D7

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 383 314 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
21.01.2004 Bulletin 2004/04

(51) Int Cl.7: **H04N 5/265**, H04N 5/275

(21) Application number: 03254060.1

(22) Date of filing: 26.06.2003

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR
Designated Extension States:
AL LT LV MK

(30) Priority: 17.07.2002 GB 0216685

(71) Applicant: SONY UNITED KINGDOM LIMITED
Weybridge KT13 0XW (GB)

(72) Inventor: Witt, Sarah Elizabeth
Basingstoke, Hampshire RG23 8WL (GB)

(74) Representative: Turner, James Arthur et al
D. Young & Co.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) **Video signal processing**

(57) A video processing method generates output images for display, each output image having one or more regions derived from a first source image separated by a wipe boundary from one or more regions derived from a second source image, each display position of the source images having an associated transparency coefficient. The method comprises the steps of: preparing the first source image for display in a display memory; defining a wipe origin locus representing a wipe boundary when substantially all of the output image is derived from the first source image and a wipe destination locus representing a wipe boundary when substantially all of the output image is derived from the second source image, points on the wipe destination locus being associated with points on the wipe origin locus, each such pair of associated points defining a respective wipe progression direction; generating a wipe control signal defining proportions of the first and second source images to be displayed with respect to normalised display distance along a wipe progression direction from the wipe origin locus and the wipe destination locus; modifying the transparency coefficient of the first source image held in the display memory, the transparency coefficient of each display position being modified in dependence on value of the wipe control signal corresponding to the normalised display distance along the wipe progression direction between that display position and the wipe origin locus and between that display position and the wipe destination locus; and writing the second source image over the first source image in the display memory so that the first source image is modified by pixels of the second source image in dependence on the transparency coefficient associated with each display position of the first source image.

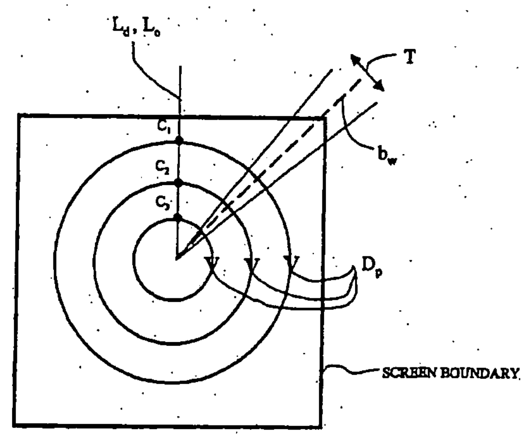

FIGURE 14

Fig. 1, 6 ; para. [0015], [0044]

EP 1 383 314 A2

generating a wipe control signal defining proportions of the first and second source images to be displayed with respect to a normalised display distance along a wipe progression direction from the wipe origin locus and the wipe destination locus;
modifying the transparency coefficient of the first source image held in the display memory, the transparency coefficient of each display position being modified in dependence on value of the wipe control signal corresponding to the normalised display distance along the wipe progression direction between that display position and the wipe origin locus and between that display position and the wipe destination locus;
writing the second source image over the first source image in the display memory so that the first source image is modified by pixels of the second source image in dependence on the transparency coefficient associated with each display position of the first source image.

[0010] Further respective aspects and features of the invention are defined in the appended claims.
[0011] Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates the overall system architecture of the PlayStation2;
Figure 2 schematically illustrates the architecture of an Emotion Engine;
Figure 3 schematically illustrates the configuration of a Graphic synthesiser;
Figure 4 is a schematic block diagram illustrating the drawing processing performed by the Graphics Synthesiser;
Figures 5A to 5G illustrate alternative primitive graphics objects used by the Graphic Synthesiser;
Figure 6 schematically illustrates a system architecture having a peripheral component interconnect (PCI) plug-in module for digital video input/output;
Figure 7 schematically illustrates a Universal Serial Bus (USB) analogue input module for importing video and audio data into the PlayStation2;
Figure 8 schematically illustrates an embodiment of the invention in which the circuitry required to implement the real-time video editing is provided on a PCI card fitted to a standard personal computer;
Figure 9 schematically illustrates a circular wipe according to an embodiment of the invention;
Figure 10 schematically illustrates a wipe control signal according to an embodiment of the invention;
Figure 11 illustrates a vertical wipe according to an embodiment of the invention;
Figure 12 illustrates a diagonal wipe according to an embodiment of the invention;
Figure 13 schematically illustrates a heart shaped wipe according to an embodiment of the invention;
Figure 14 schematically illustrates a clock wipe according to an embodiment of the invention;
Figure 15A schematically illustrates a portion of texture space;
Figure 15B schematically illustrates a graphics primitive in co-ordinate space;
Figure 15C schematically illustrates a wipe control signal which is applied as a one dimensional texture function to the graphics primitive; and
Figure 16 is a flow chart explaining how a wipe is performed according to embodiments of the invention.

[0012] Figure 1 schematically illustrates the overall system architecture of the PlayStation2. The system comprises: an Emotion Engine 100; a Graphics Synthesiser 200; a sound processor unit 300 having dynamic random access memory (DRAM); a read only memory (ROM) 400; a compact disc (CD) and digital versatile disc (DVD) unit 450; a Rambus Dynamic Random Access Memory (RDRAM) unit 500; an input/output processor 700 with dedicated RAM 750; and an external hard disk drive (HDD) 800.
[0013] The Emotion Engine 100 is a 128-bit Central Processing Unit (CPU) that has been specifically designed for efficient simulation of 3 dimensional (3D) graphics for games applications. The Emotion Engine components include a data bus, cache memory and registers, all of which are 128-bit. This facilitates fast processing of large volumes of multi-media data. Conventional PCs, by way of comparison, have a basic 64-bit data structure. The floating point calculation performance of the PlayStation2 is 6.2 GFLOPs. The Emotion Engine also comprises MPEG2 decoder circuitry which allows for simultaneous processing of 3D graphics data and DVD data. The Emotion Engine performs geometrical calculations including mathematical transforms and translations and also performs calculations associated with the physics of simulation objects, for example, calculation of friction between two objects. It produces sequences of image rendering commands which are subsequently utilised by the Graphics Synthesiser 200. The image rendering commands are output in the form of display lists. A display list is a sequence of drawing commands that specifies to the Graphics Synthesiser which primitive graphic objects (e.g. points, lines, triangles, sprites) to draw on the screen and at which co-ordinates. Thus a typical display list will comprise commands to draw vertices, commands to shade the faces of polygons, render bitmaps and so on. The Emotion Engine 100 can asynchronously generate multiple display lists.
[0014] The Graphics Synthesiser 200 is a video accelerator that performs rendering of the display lists produced by

[0023] Vector unit one 108 comprises 5 FMACS and 2 FDIVs. It has no direct path to the CPU core 102, although it does have a direct path to the GIF unit 110. It has 16 Kb of vector unit memory for data and 16 Kb of micro-memory for instructions. Vector unit one 108 is useful for performing transformations. It primarily executes patterned geometric processing and directly outputs a generated display list to the GIF 110.

[0024] The GIF 110 is an interface unit to the Graphics Synthesiser 200. It converts data according to a tag specification at the beginning of a display list packet and transfers drawing commands to the Graphics Synthesiser 200 whilst mutually arbitrating multiple transfer. The interrupt controller (INTC) 112 serves to arbitrate interrupts from peripheral devices, except the DMAC 116.

[0025] The timer unit 114 comprises four independent timers with 16-bit counters. The timers are driven either by the bus clock (at 1/16 or 1/256 intervals) or via an external clock. The DMAC 116 handles data transfers between main memory and peripheral processors or main memory and the scratch pad memory. It arbitrates the main bus 124 at the same time. Performance optimisation of the DMAC 116 is a key way by which to improve Emotion Engine performance. The image processing unit (IPU) 118 is an image data processor that is used to expand compressed animations and texture images. It performs I-PICTURE Macro-Block decoding, colour space conversion and vector quantisation. Finally, the sub-bus interface (SIF) 122 is an interface unit to the IOP 700. It has its own memory and bus to control I/O devices such as sound chips and storage devices.

[0026] Figure 3 schematically illustrates the configuration of the Graphic Synthesiser 200. The Graphics Synthesiser comprises: a host interface 202; a set-up / rasterizing unit 204; a pixel pipeline 206; a memory interface 208; a local memory 212 including a frame page buffer 214 and a texture page buffer 216; and a video converter 210.

[0027] The host interface 202 transfers data with the host (in this case the CPU core 102 of the Emotion Engine 100). Both drawing data and buffer data from the host pass through this interface. The output from the host interface 202 is supplied to the graphics synthesiser 200 which develops the graphics to draw pixels based on vertex information received from the Emotion Engine 100, and calculates information such as RGBA value, depth value (i.e. Z-value), texture value and fog value for each pixel. The RGBA value specifies the red, green, blue (RGB) colour components and the A (Alpha) component represents opacity of an image object. The Alpha value can range from completely transparent to totally opaque. The pixel data is supplied to the pixel pipeline 206 which performs processes such as texture mapping, fogging and Alpha-blending (as explained below) and determines the final drawing colour based on the calculated pixel information.

[0028] The pixel pipeline 206 comprises 16 pixel engines PE1, PE2 .. PE16 so that it can process a maximum of 16 pixels concurrently. The pixel pipeline 206 runs at 150MHz with 32-bit colour and a 32-bit Z-buffer. The memory interface 208 reads data from and writes data to the local Graphics Synthesiser memory 212. It writes the drawing pixel values (RGBA and Z) to memory at the end of a pixel operation and reads the pixel values of the frame buffer 214 from memory. These pixel values read from the frame buffer 214 are used for pixel test or Alpha-blending. The memory interface 208 also reads from local memory 212 the RGBA values for the current contents of the frame buffer. The local memory 212 is a 32 Mbit (4MB) memory that is built-in to the Graphics Synthesiser 200. It can be organised as a frame buffer 214, texture buffer 216 and a 32-bit Z-buffer 215. The frame buffer 214 is the portion of video memory where pixel data such as colour information is stored.

[0029] The Graphics Synthesiser uses a 2D to 3D texture mapping process to add visual detail to 3D geometry. Each texture may be wrapped around a 3D image object and is stretched and skewed to give a 3D graphical effect. The texture buffer is used to store the texture information for image objects. The Z-buffer 215 (also known as depth buffer) is the memory available to store the depth information for a pixel. Images are constructed from basic building blocks known as graphics primitives or polygons. When a polygon is rendered with Z-buffering, the depth value of each of its pixels is compared with the corresponding value stored in the Z-buffer. If the value stored in the Z-buffer is greater than or equal to the depth of the new pixel value then this pixel is determined visible so that it should be rendered and the Z-buffer will be updated with the new pixel depth. If however the Z-buffer depth value is less than the new pixel depth value the new pixel value is behind what has already been drawn and will not be rendered.

[0030] The local memory 212 has a 1024-bit read port and a 1024-bit write port for accessing the frame buffer and Z-buffer and a 512-bit port for texture reading. The video converter 210 is operable to display the contents of the frame memory in a specified output format.

[0031] Figure 4 is a schematic block diagram illustrating the drawing processing performed by the Graphics Synthesiser 200. The drawing processing modules comprise: a texture mapping module 252; a fogging module 254; an anti-aliasing module 256; a pixel test module 258; an alpha blending module 260; and a formatting module 262.

[0032] As explained above, the Graphics Synthesiser 200 receives display lists from the Emotion Engine 100. Each display list is pre-processed such that a gradient (e.g. shading coefficient) and other parameters appropriate for drawing graphics primitives are calculated based on vertex information contained in the display list. The pixels of a graphics primitive are generated by a Digital Differential Algorithm (DDA) during a process known as rasterizing. This rasterizing process involves concurrently generating values for 8 or 16 pixels. Essentially a 3D image is transformed into a set of coloured pixels and the colour assigned to each pixel will depend on light sources, the position of the object that the

frame memory.

[0039] Following processing by the Alpha Blending Module 260, the data is supplied to the formatting module 262 where the pixel values for drawing are converted to the data format of the frame buffer. Dithering and colour clamping may also be applied at this stage. The dithering process involves creating a new colour by blending several colours which are already available. This technique can be used to give the illusion that an image was rendered with 64K colours although it was actually rendered with 256 colours. Colour clamping is a process whereby the RGB value of a pixel is controlled to be within the range 0-255 (8-bit value). Since the value of a pixel occasionally exceeds this range after operations such as Alpha-blending the result is stored with 9-bits for each RGB value.

[0040] Output from the data formatting module is supplied to the memory interface 208 (Figure 3) via which read/write is performed to local memory 212. The operations supported by the memory interface include: writing drawing pixel values RGBA and Z to memory following a pixel operation; reading pixel values into the frame buffer e.g. during pixel test and alpha-blending processes; and reading RGBA values from memory for display on the screen.

[0041] Figures 5A to 5G illustrate alternative primitive graphics objects used by the Graphic Synthesiser 200. The alternative graphics primitives comprise a Point, a Line, a LineStrip, a Triangle, a TriangleStrip, a TriangleFan and a Sprite. Figure 5A illustrates three independent points, each of which is drawn with a single piece of vertex information. Figure 5B shows two independent Lines, each of which is drawn with 2 pieces of vertex information. Figure 5C illustrates a LineStrip comprising 4 lines which share endpoints. In this case the first line is drawn with two pieces of vertex information whereas succeeding lines are drawn with a single piece of vertex information. Figure 5D shows two independent Triangles, each of which is drawn using 3 pieces of vertex information. Figure 5E illustrates a TriangleStrip comprising 5 triangles which are continuous in that they share sides. In this case the first triangle is drawn using 3 pieces of vertex information and each succeeding triangles is drawn whenever a single piece of vertex information is added. Figure 5F shows a TriangleFan comprising 5 triangles which share a common vertex 1. The first triangle requires 3 pieces of vertex information whereas succeeding triangles are drawn whenever a single piece of vertex information is added. Figure 5G shows two independent rectangles known as Sprites. Each Sprite is drawn using 2 pieces of vertex information representing diagonally opposite corners of the rectangle.

[0042] The general drawing procedure performed by the Graphics Synthesiser 200 involves: firstly setting the primitive type and initialising the condition of a vertex queue; secondly setting vertex information including drawing co-ordinates, vertex colour, texture co-ordinates and fog coefficients, in vertex information setting registers; thirdly performing a "vertex kick" operation whereby vertex information set up to this point is placed in the vertex queue and the queue goes one step forward; and finally, when the appropriate vertex information is arranged in the vertex queue, commencing the drawing process.

[0043] It is appropriate to provide an interface in order to import video and audio data into the PlayStation2. The HDD 900 requires video data in MPEG2 I-frame only format and audio data in PCM format so that hardware is required to convert either DV streams or analogue video/audio into the format required by the HDD. Hardware must also be provided to allow the output video and audio to be converted back to DV format so that it can be digitally recorded by the user.

[0044] Figure 6 schematically illustrates a system architecture having a peripheral component interconnect (PCI) plug-in module for digital video input/output. This apparatus comprises the Sound Processor Unit 300, the IOP 700, the Emotion Engine 100 and the Graphics Synthesiser 200 of the PlayStation2 main unit as described above with reference to Figure 1. The apparatus of Figure 6 also comprises a PCI interface 902 to which additional hardware module 904 comprising a hard disc drive (HDD) 906 and a DV/MPEG2 plug-in module 908 is connected. The IOP 700 is provided with 2 USB ports, 2 controller ports and 2 memory card ports and a full-speed 400Mbps IEEE 1394 (iLink) port 702. DV video is a compression standard for camcorders and video tape recorders. DV format data is stored in binary format rather than analogue format. MPEG2 is a standard developed by the Moving Pictures Expert Group. It is a digital encoding technology capable of encoding a video plus audio bitstream at variable encoding rates up to 15Mbits/s, with the video occupying up to 9.8Mbit/s. MPEG2 encoding is used on DVDs.

[0045] The so-called "iLink" is the Sony Corporation implementation of the IEEE1394 High Performance serial Bus standard. This standard describes a serial bus or pathway between one or more peripheral devices and a microprocessor device. The iLink provides a single plug-and-socket connection on which up to 63 peripheral devices can be attached. The iLink port 702 of the IOP 700 can be used to import DV video which is routed through to the DV/MPEG2 plug-in module 908 that is attached to the PCI port 902. Using this apparatus output video may be converted from MPEG2 to DV and output through the iLink port 702. To facilitate input of analogue input video/audio data (such as S-Video or Composite video and stereo audio) additional connectors (not shown) must be inserted in the hardware module 904.

[0046] The DV/MPEG2 module 908 is used to convert input video data in DV format to MPEG2 video and Pulse Code Modulated (PCM) audio which is then stored on the HDD 906 on input. At the output stage the hardware module 904 may be used to convert output video and audio into DV format which is output via the iLink port 702.

[0047] Figure 7 schematically illustrates a Universal Serial Bus (USB) analogue input module for importing video

the wipe boundary bw is evolved by progressively magnifying the shape formed by the wipe origin locus, as if projecting an image of the current wipe boundary onto a more distant focal plane.

[0055] In the example embodiment of Figure 9, there is a 50:50 mix between source images S_1 and S_2 for display pixels coinciding with the wipe boundary. Moving away from any given point on the wipe boundary along the wipe progression direction Dp (which in this case is an outward radial direction passing through the given point), the proportion of S_1 progressively increases within the transition region up to 100% at the transition region outer boundary radius r_o. Moving in the opposite direction, away from the wipe boundary and towards the wipe origin, the proportion of S_2 gradually increases until reaching 100% at the transition region inner boundary radius r_i. Consider for example moving from the point B on the wipe boundary bw, along the associated wipe progression direction OP.

[0056] In order to perform the wipe it is appropriate to define the wipe geometry, the boundaries of the transition region r_i and r_o and the functional form of the mix in the transition region. A discrete set of points forming the wipe destination locus and the wipe origin locus is either generated in real-time or retrieved from memory in response to user-selection from a menu of available wipe geometries. As shown in Figure 9, the wipe destination locus Ld is defined to be outside the screen area to ensure that the 100% wipe stage corresponds to the screen being completely filled by S_2.

[0057] Figure 10 schematically illustrates a wipe control signal according to an embodiment of the invention. In this case a single parameter, the normalised distance of a given display position from the wipe origin measured along the wipe progression direction is determined and the functional form of the wipe control signal illustrated in Figure 10 is used to specify the relative proportions of each source image to be displayed at that display position in dependence upon the determined distance.

[0058] For the circular wipe geometry of Figure 9, the relative proportions of the source images S_1 and S_2 at a given display position could be consistently defined in terms of the absolute distance along the wipe progression direction. It can be seen from the figure that along the lines OP and OQ, the same mix proportion corresponds to the same absolute distance. As explained below with reference to Figure 13, the same rule does not apply more irregular wipe geometries.

[0059] Consider taking a sample of pixel values along the line OP in Figure 9. The wipe control signal of Figure 10 defines the source image composition of the displayed image as a function of distance x along the line OP. Accordingly the wipe control signal has zero gradient and a constant y-value y_2 (associated with S_2) in the region $O \leq x \leq r_i$, which corresponds to the inner region where the displayed image is derived from S_2 only. Similarly the wipe control signal has zero gradient and a different constant y-value y_1 (associated with S_1) in the region $r_o \leq x \leq P$ which corresponds to the outer region where the displayed image is derived from S_1 only. However in the region $r_i < x < r_o$, the wipe control signal smoothly varies from the first constant y-value to the second constant y-value and defines the relative proportions of the image derived from S1 and S2 as the transition region is traversed. For example the mix in the transition region could be defined by $\alpha a y_1 + (1-\alpha)\, y_2$ where a is a variable that satisfies boundary conditions $\alpha=0$ at $x=r_i$; $\alpha=1$ at $x=r_o$ and $\alpha=0.5$ at the wipe boundary. Here, y_i is an RGB value corresponding to source image S_i.

[0060] The functional form of the wipe control signal is not by any means restricted to the functional form shown in Figure 10. In alternative embodiments the wipe control signal may be characterised, for example by a sinusoidal curve in the transition region and although the gradient must be zero for the inner and outer image regions, the values y_1 and y_2 can be appropriately selected. As will be explained below, the y value of the wipe control signal is used to replace in the frame memory, the alpha channel values of pixels of the first source image S_1 during implementation of the wipe.

[0061] As explained above, the prior art system implements the circular wipe using a 3-dimensional wipe function representing a right circular cone and the transition region boundaries r_i and r_o are defined via upper and lower cone height thresholds. The cone height value for each pixel depends upon the radius of that pixel from the wipe origin O and this ultimately determines whether that pixel belongs to the inner image region, the outer image region or the transition region. By way of contrast, this embodiment of the invention uses a 1-dimensional wipe control signal rather than a 3-dimensional cone function to define the relative proportions of the two source images displayed at each pixel. The wipe control signal is used to modify the alpha channel of the first source image in screen memory. The second source image is subsequently applied according to a mix defined by the modified alpha channel.

[0062] Figure 11A schematically illustrates a vertical wipe according to an embodiment of the invention. In this case the second source image S_2 progressively replaces the original source image S1 as the wipe progresses. The wipe boundary traces out a path starting from a wipe origin locus at the right hand edge of the screen and finishing at a wipe destination locus at the far left-hand edge of the screen. In this case the vertical line Lo is the wipe origin locus and a parallel line (not shown) at the left hand edge of the screen is the wipe destination locus. A nominal wipe boundary b_w is defined in the centre of the transition region and parallel to the wipe origin and wipe destination loci. In this case, for a given display point P_3, the relative proportions of S_1 and S_2 in the displayed image is defined by the wipe control signal. The x co-ordinate of the wipe control signal is given by the normalised distance U_3P_3/U_3W_3 along the wipe progression direction defined by the vector U_3W_3. Figure 11B shows the value of the wipe control signal corresponding

the vertices of the graphics primitive in the display "co-ordinate space" which in this implementation has full-screen dimensions of 720 by 576, a standard resolution for PAL systems. Scaling and rotation of the texture may also need to be performed to ensure that its size and orientation correctly match the graphics primitive to which it is applied.

[0071] As illustrated in Figure 15C, the wipe control signal is applied to the graphics primitive as a one-dimensional texture function. The wipe control signal is used to modify only the alpha channel A of the RGBA video parameters. In this case the alpha value specified by the ID texture function corresponding to normalised distance = 0 is applied to all pixels along the line AD of the graphics primitive whilst the alpha value specified by the 1D texture function corresponding to normalised distance = 1 is applied to all pixels along the line BC of the graphics primitive. Thus the alpha value varies within the graphics primitive in accordance with the 1D texture function in a direction parallel to AB. In this example the graphics primitive must be straddling the transition region of the wipe. In the case of the circular wipe of Figure 9, the alpha values of all graphics primitives in the inner region are fixed at a low value corresponding to a high degree of transparency whereas the alpha values of the graphics primitives in the outer region would be set to a high value corresponding to an opaque image. In the transition region, the alpha values of the image pixels will vary in accordance with the wipe control signal along the wipe progression direction i.e. the 1D texture function is applied along the wipe progression direction.

[0072] The 1D texture function for each graphics primitive is derived from the global wipe control signal via the normalised distance to an associated display point. However the alpha values of the displayed image are modified pixel by pixel in accordance with the appropriate 1D texture function (defined via mapping from texture co-ordinates to screen co-ordinates). The wipe control signal is used to determine alpha values for pixels along a direction on the screen corresponding the wipe progression direction.

[0073] The alpha values $A_{texture}$ specified by the wipe control signal are applied as 1D textures to the graphics primitives and are used to replace the alpha channel A_{S1} values of the first source video S_1. The alpha channel transformation (A_{S1}, R_{S1}, G_{S1}, B_{S1}) --> ($A_{texture}$, R_{S1}, G_{S1}, B_{S1}) is performed by modifying values in the frame buffer memory only, so there is no change to the video output at this stage. Once the alpha channel of the first source video S_1 has been replaced, the second source video S_2 is applied using the alpha channel values from the frame buffer memory $A_{texture}$ to define the mix. Thus:

$$OutputColour = A_{texture} * S_1Colour + (1 - A_{texture}) * S_2Colour$$

[0074] Where $S_1Colour$ is R_{S1}, G_{S1} or B_{S1} ; $S_2Colour$ is R S_2, G_{S2} or B_{S2}; and *OutputColour* is the displayed colour resulting from the mix. The above formula assumes that $A_{texture}$ has been normalised.

[0075] For example for the circular wipe, replacement of the alpha channel of S_1 by values $A_{texture}$, specified by the wipe control signal of Figure 10, will result in the inner region having a high degree of transparency, the transition region having a degree of transparency that progressively decreases from r_1 to r_2 and an outer region that is opaque. Thus when the second source video S_2 is applied and a mix is performed according to the above formula, the high degree of transparency of S_2 in the inner region will mean that the image visible there corresponds to S_2 whereas the opacity of S_1 in the outer region means that S_2 will not be visible there. In the transition region, S_1 is at least partially transparent so that the displayed image there has a partial contribution from the pixel values of both S_1 and S_2, the relative proportions having been defined via the wipe control function.

[0076] Figure 16 is a flow chart that schematically illustrates how a wipe is performed according to embodiments of the invention. At step 1410 the emotion engine 100 generates a wipe control signal. At step 1420 the emotion engine calculates the texture co-ordinates and maps them to the appropriate screen co-ordinates. At stage 1430 the wipe control signal is downloaded from the emotion engine to the graphics synthesiser 200. At stage 1440 the graphics synthesiser applies the wipe control signal to the graphics primitives of the image by creating a 1D texture function for each graphics primitive. The 1D texture function is derived from the wipe control signal in dependence upon the location and orientation of the graphics primitive in relation to the wipe edge. At stage 1450 the result of applying the 1D texture functions to the graphics primitives are that the alpha channel A_{S1} of the first source video S_1 is replaced by the value $A_{texture}$ specified by the 1D texture function. At stage 1460 the second source video S_2 is applied using alpha channel values $A_{texture}$ from screen memory. The result of the mix is displayed on the screen and has the effect of implementing the wipe at a particular wipe progression stage. The form of the wipe control signal changes as the wipe progresses, tracking the movement of the wipe boundary on the display screen

[0077] It will be appreciated from the above that the invention may be implemented as computer software, which may be supplied on a storage medium or via a transmission medium such as a network or the Internet.

means for defining a wipe origin locus representing a wipe boundary when substantially all of the output image is derived from the first source image and a wipe destination locus representing a wipe boundary when substantially all of the output image is derived from the second source image, points on the wipe destination locus being associated with points on the wipe origin locus, each such pair of associated points defining a respective wipe progression direction;
means for generating a wipe control signal defining proportions of the first and second source images to be displayed with respect to a normalised display distance along a wipe progression direction from the wipe origin locus and the wipe destination locus;
means for modifying the transparency coefficient of the first source image held in the display memory, the transparency coefficient of each display position being modified in dependence on value of the wipe control signal corresponding to the normalised display distance along the wipe progression direction between that display position and the wipe origin locus and between that display position and the wipe destination locus; and
means for writing the second source image over the first source image in the display memory so that the first source image is modified by pixels of the second source image in dependence on the transparency coefficient associated with each display position of the first source image.

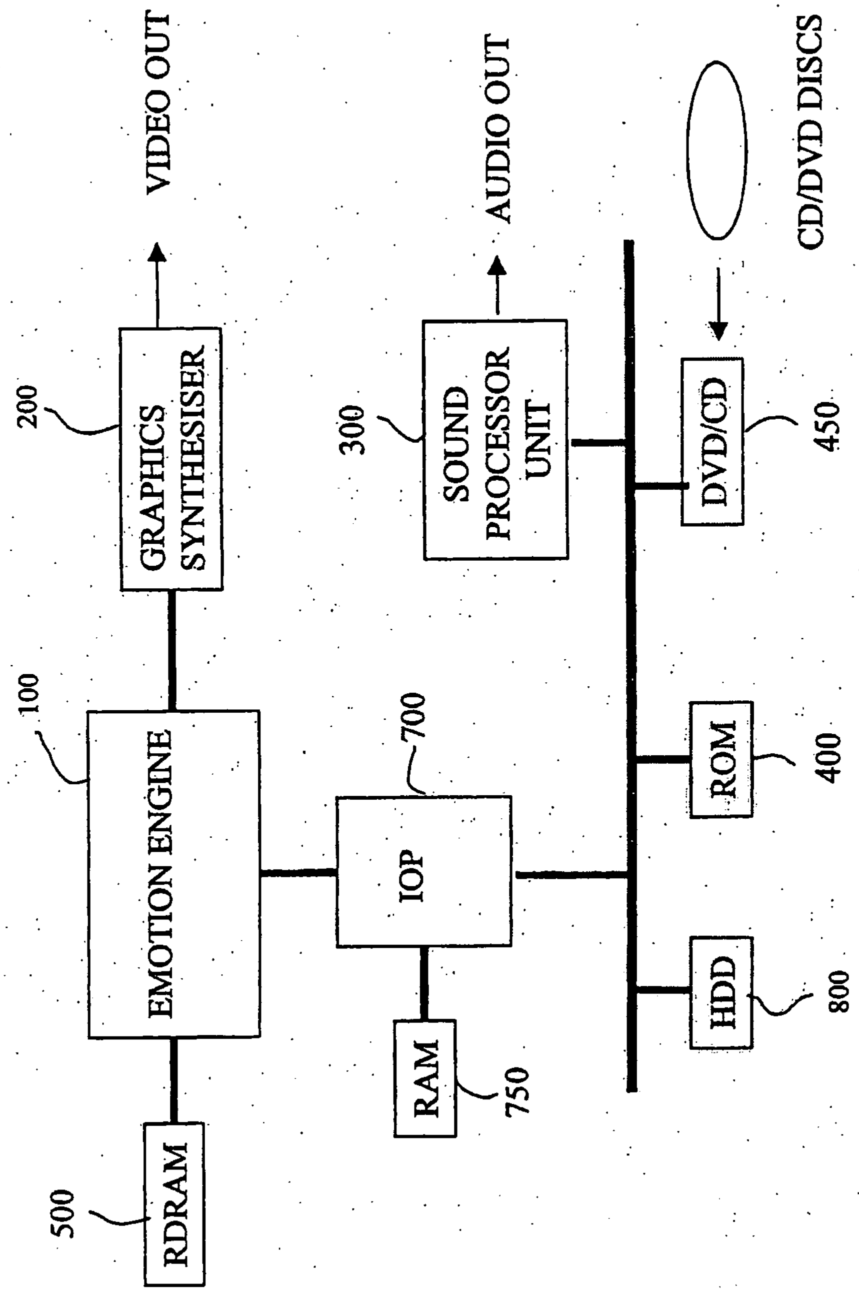
VIDEO OUT
AUDIO OUT
CD/DVD DISCS
200
GRAPHICS SYNTHESISER
300
SOUND PROCESSOR UNIT
DVD/CD
450
100
EMOTION ENGINE
700
IOP
ROM
400
HDD
800
RAM
750
RDRAM
500

FIGURE 1

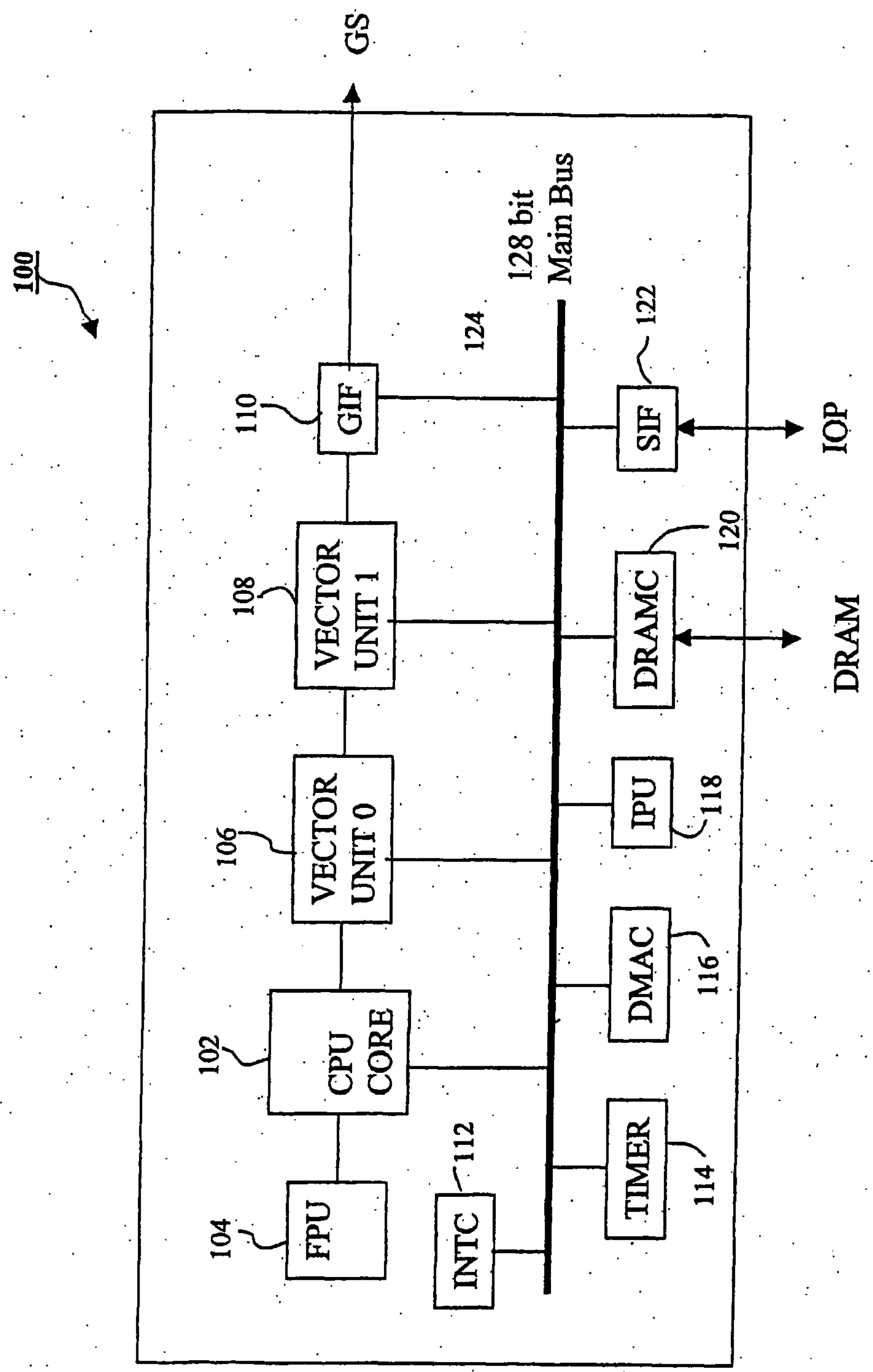
100
GS
128 bit
Main Bus
124
110
GIF
122
SIF
IOP
108
VECTOR
UNIT 1
120
DRAMC
DRAM
106
VECTOR
UNIT 0
IPU
118
DMAC
116
102
CPU
CORE
104
FPU
112
INTC
TIMER
114

FIGURE 2

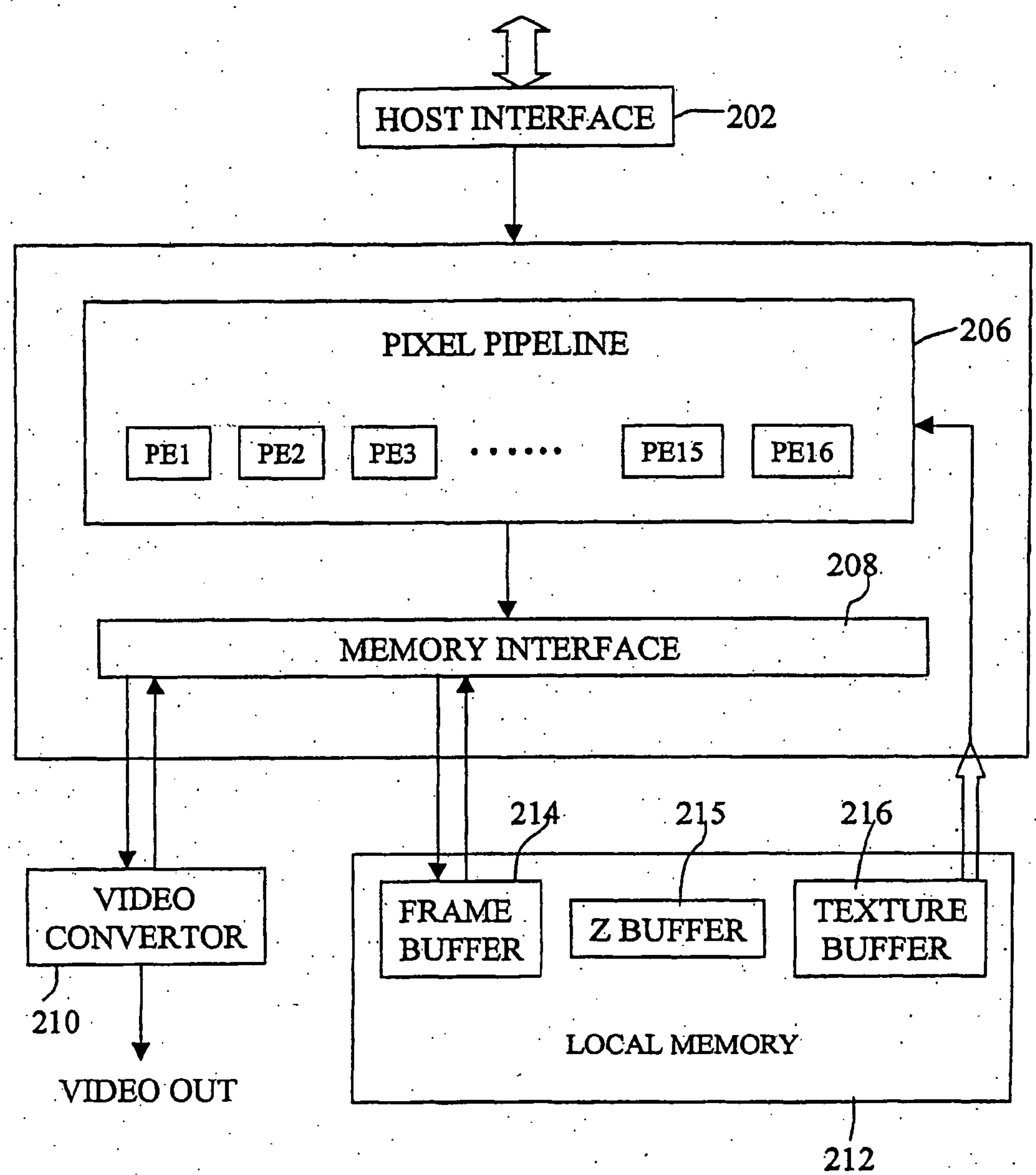
HOST INTERFACE
202
PIXEL PIPELINE
206
PE1
PE2
PE3
PE15
PE16
208
MEMORY INTERFACE
214
215
216
VIDEO CONVERTOR
FRAME BUFFER
Z BUFFER
TEXTURE BUFFER
210
LOCAL MEMORY
VIDEO OUT
212

FIGURE 3

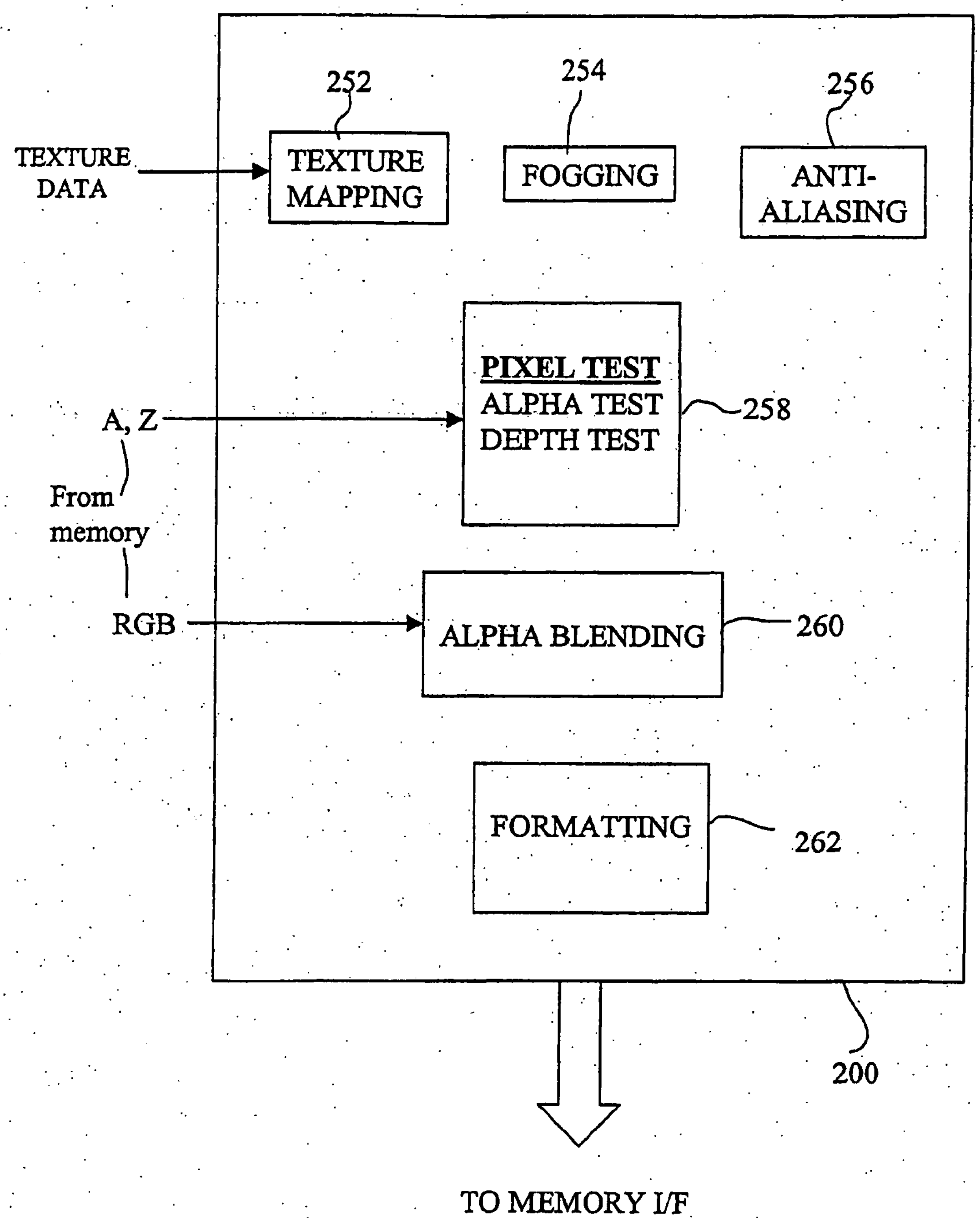
252
254
256
TEXTURE DATA
TEXTURE MAPPING
FOGGING
ANTI-ALIASING
PIXEL TEST
ALPHA TEST
DEPTH TEST
258
A, Z
From memory
RGB
ALPHA BLENDING
260
FORMATTING
262
200
TO MEMORY I/F

FIGURE 4

FIG 5A

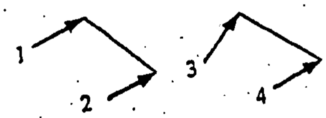

FIG 5B

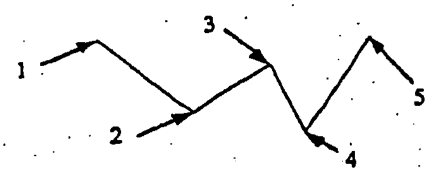

FIG 5C

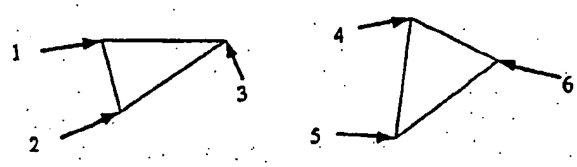

FIG 5D

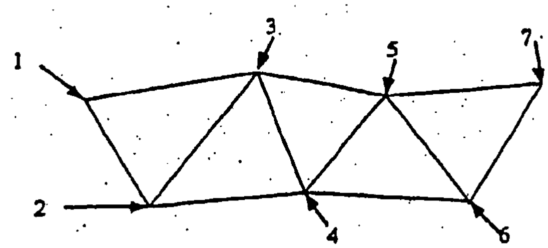

FIG 5E

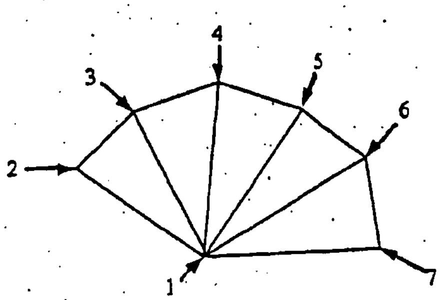

FIG 5F

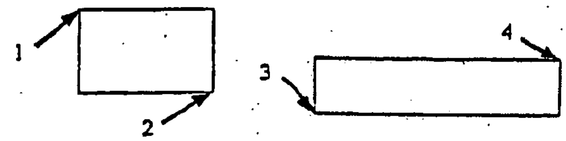

FIG 5G

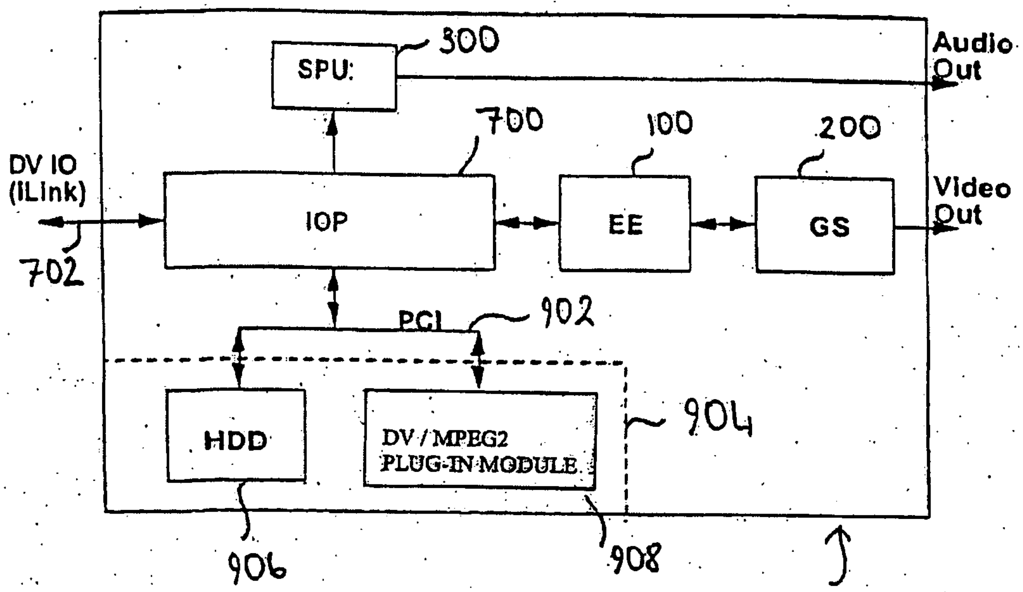

FIGURE 6

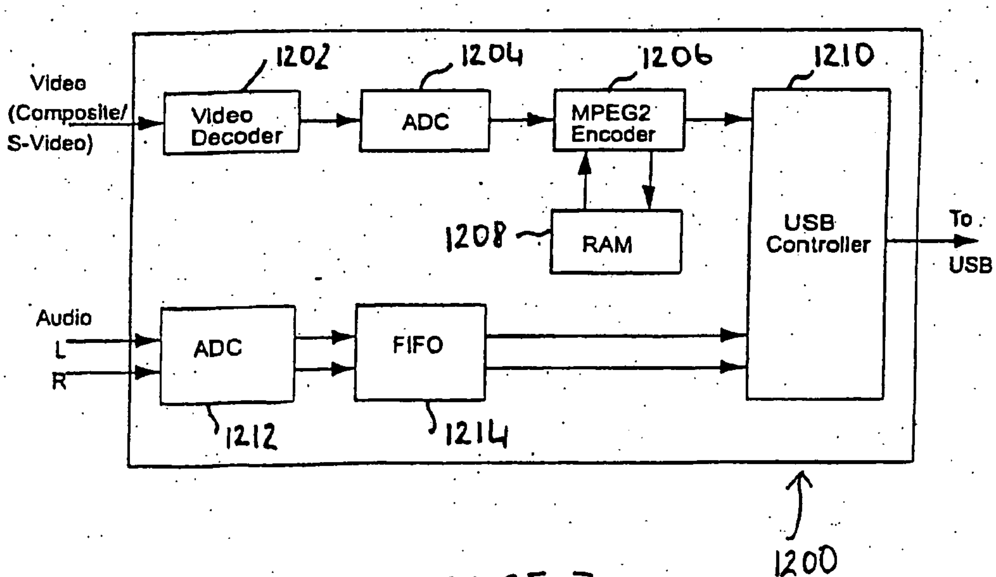

FIGURE 7

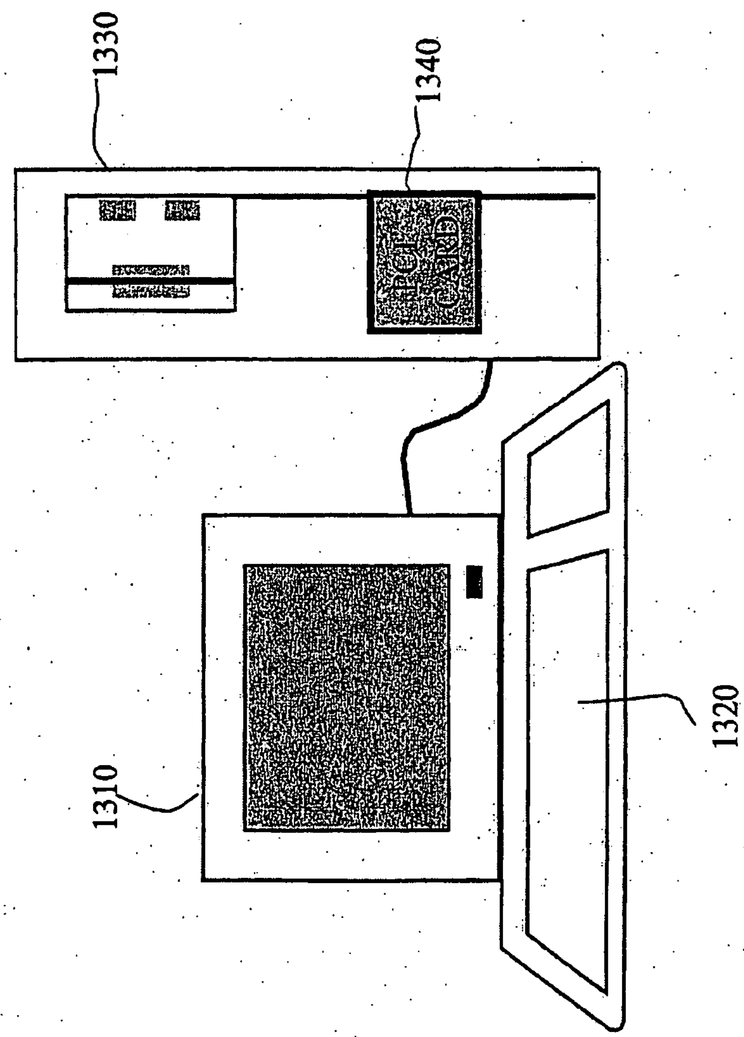

FIGURE 8

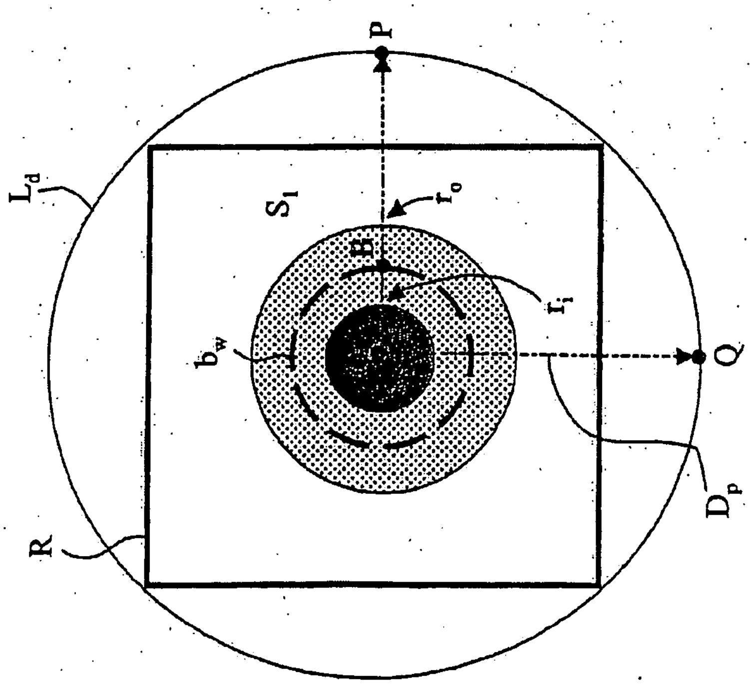

FIGURE 9

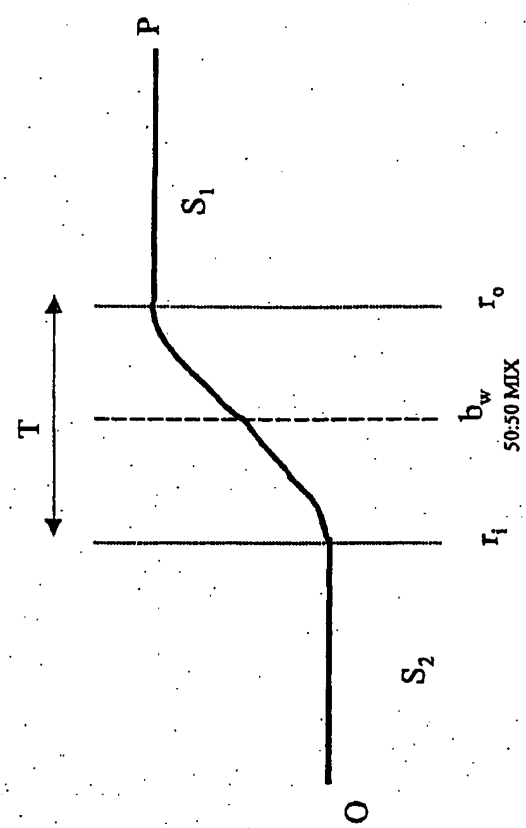

FIGURE 10

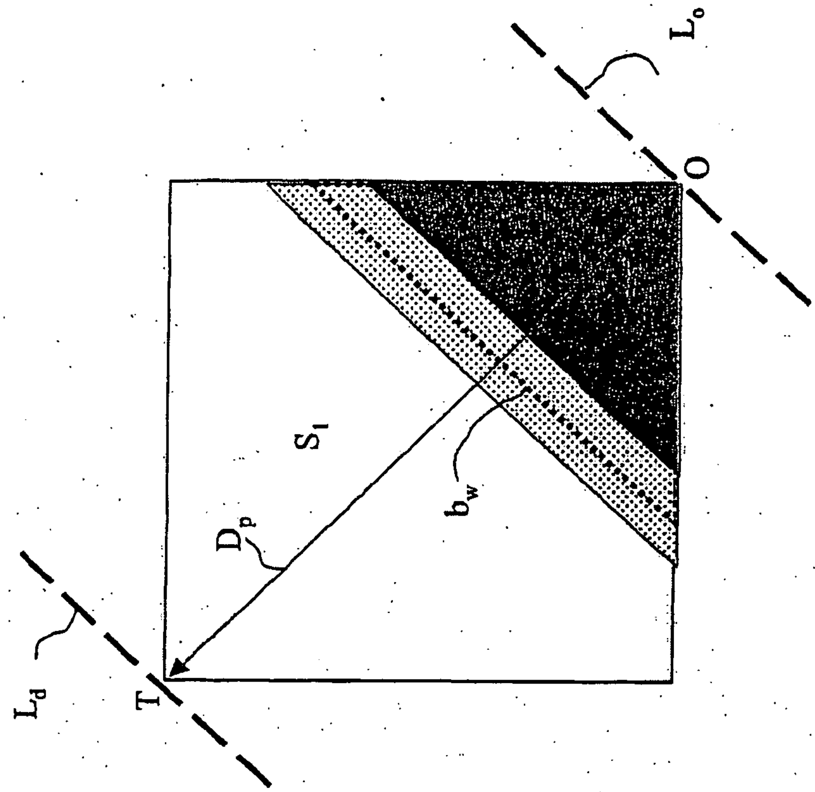

FIGURE 12

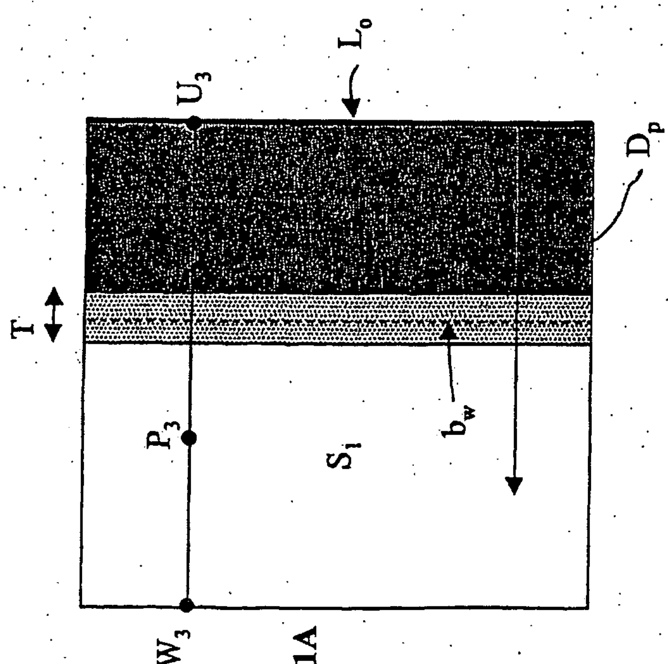

FIG 11A

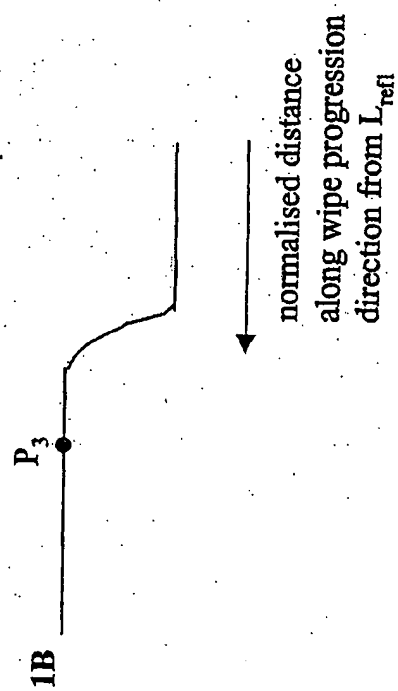

FIG 11B

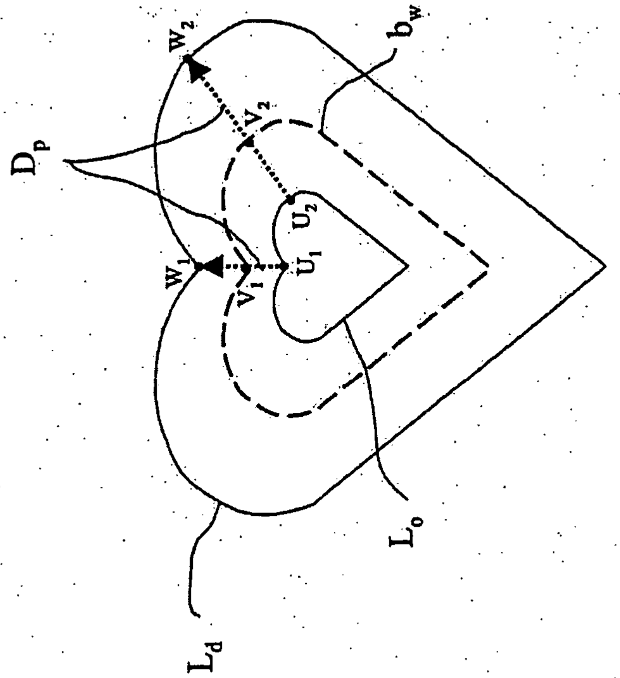

FIGURE 13

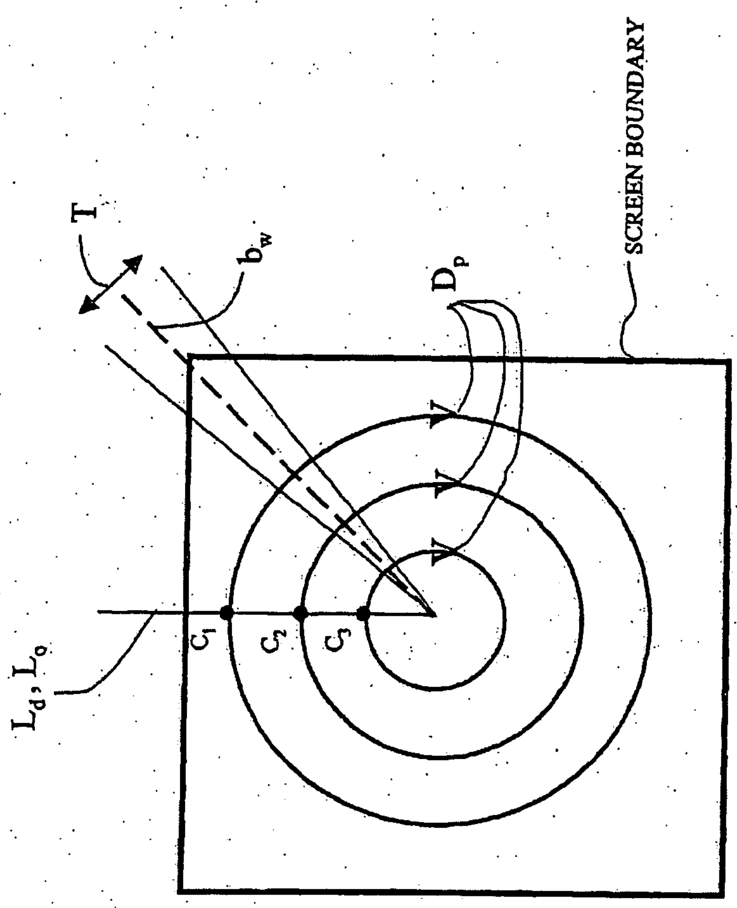

FIGURE 14

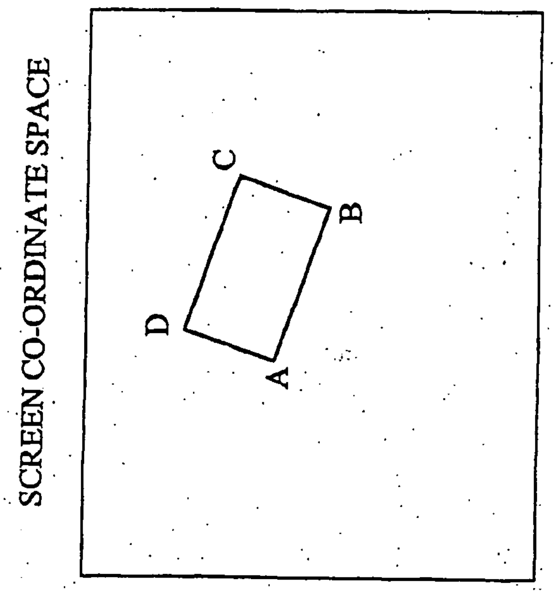

FIG 15B

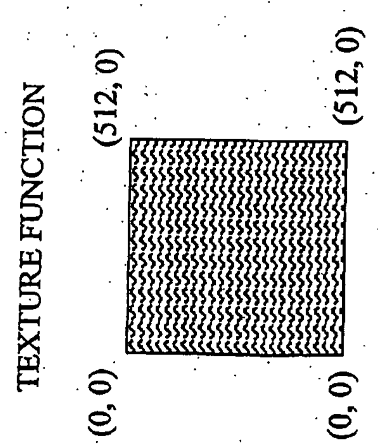

FIG 15A

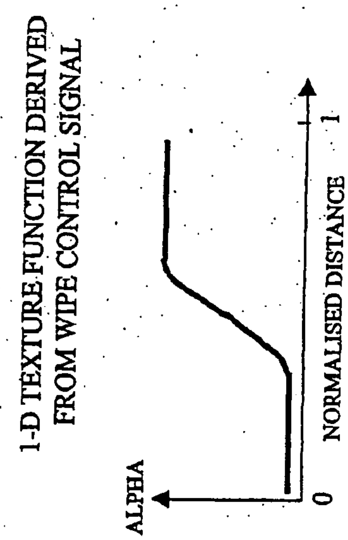

FIG 15C

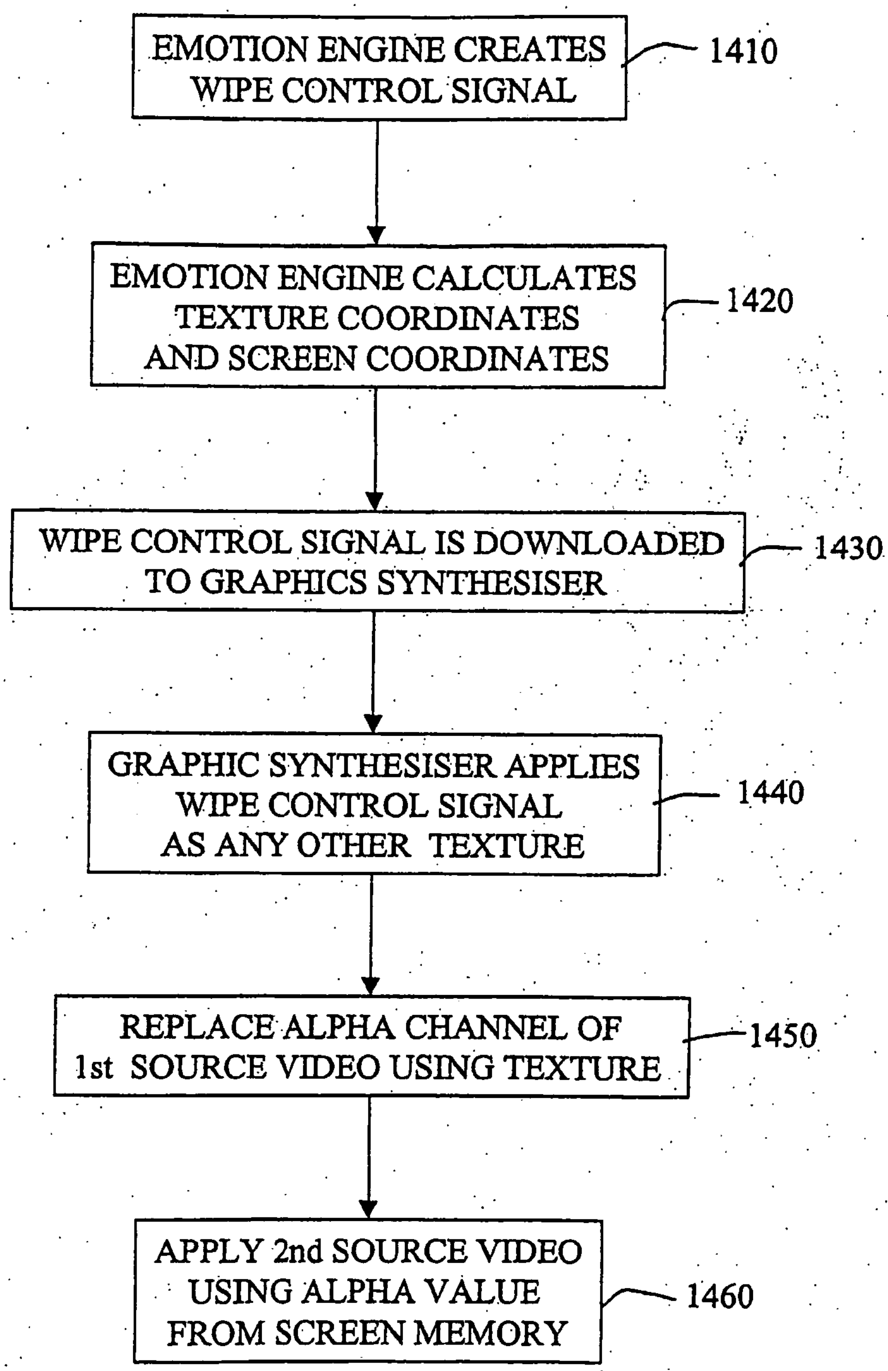
EMOTION ENGINE CREATES WIPE CONTROL SIGNAL
1410
EMOTION ENGINE CALCULATES TEXTURE COORDINATES AND SCREEN COORDINATES
1420
WIPE CONTROL SIGNAL IS DOWNLOADED TO GRAPHICS SYNTHESISER
1430
GRAPHIC SYNTHESISER APPLIES WIPE CONTROL SIGNAL AS ANY OTHER TEXTURE
1440
REPLACE ALPHA CHANNEL OF 1st SOURCE VIDEO USING TEXTURE
1450
APPLY 2nd SOURCE VIDEO USING ALPHA VALUE FROM SCREEN MEMORY
1460

FIGURE 16